Compiling Regular Expressions to Extract Legal Modifications

Compiling Regular Expressions to Extract Legal Modifications

Livio Robaldo & Leonardo Lesmo & Daniele P. Radicioni
[a] *Dipartimento di Informatica, Università di Torino*
{robaldo,lesmo,radicion}@di.unito.it

Abstract.
In this paper we present a prototype for automatically identifying and classifying types of modifications in Italian legal text. The prototype is part of the Eunomos system, a legal knowledge management service that integrates and makes available legislation from various sources, while finding definitions and explanations of legal concepts in a given context. The design of the prototype is grounded on the error analysis of a previous prototype. The latter made use of dependency relations provided by the TUP parser, a multi-purpose parser for Italian. Since those syntactic relations were responsible of the majority of errors, we decided in the present tool to ignore them, and to rewrite an ad-hoc shallow parsing, based on the morphological analysis of the legal text (still provided by the TUP parser). We obtained performances much greater than those of the initial prototype. In particular, the level of precision of the classification in output is now close to 100%.

Keywords. Shallow Parsing, Eunomos, NormeInRete

1. Introduction

A huge amount of documents is being produced that demands for intelligent approaches to be undertaken, with many purposes. Automatic indexing, querying, searching, filtering, retrieving and annotating text documents is thus a major challenge to access information in such a setting. We are presently concerned with Italian legal texts: in this domain a variant of the mentioned problem exists, in that legal documents (be them laws or other sorts of legal documents) are being produced at such a rate that tools that assist human experts in handling documents are becoming more and more necessary.

Some approaches have been proposed to build systems for automatically identifying and classifying structural portions of legal documents [1] and their intra- and inter-references [10]. Other researches are being carried out to produce semantic analysis [12,13,8]. Such interest is witnessed by various initiatives at the national and international levels, that have established XML standards for describing legal sources and schemas to identify legal documents [7]. Since the annotation process is expensive and error-prone, such efforts will not be really useful unless are available tools to extract in automatic (or supervised) fashion both *structural* and *semantic* data from legal texts. For instance, an editor based on the Italian standard NormeInRete (NIR) [1] allows using a text editor to mark up in an automated or semi-supervised fashion structural partitions and normative

references. In contrast, the task of automatic annotation of semantic data is still an open issue.

In this paper we consider the problem of the annotation of *modificatory provisions*. A modificatory provision is a change made to one or more clauses within a text, to the entire text, or to the relations that hold among the constituent provisions of a legal system (as when a decree-law is made into law). Modificatory provisions are particularly relevant, since they affect the whole normative system. It should be considered, in this regard, that a lavish use of normative modifications tends to undermine the certainty of the law, so that the changes are sometimes fragmentary and incoherent, making it even more difficult to clearly understand what is the law, or which one of several versions of a provision counts as law.
The research presented in this paper is collocated within the development of the Eunomos system [2], whose goals and capabilities are briefly outlined below.

In this paper we describe a NLP system that combines shallow syntactic analysis and shallow semantic interpretation of natural language in order to enhance the NIR annotation with semantic meta-data. The system relies on some modules that are part of the Turin University Parser (TUP, [5]) that is used to generate a flat structure of the sentences with annotated POS. A rule-based semantic interpreter implements a pattern matching strategy based on regular expressions. From a theoretical point of view, this work explores in how far the undertaken approach is appropriate. On the one hand we exploit the output of the morphological analyzer of the parser (which is very robust); on the other hand the approach suffers from known limitations (e.g., it is hard to reap long span dependences). From a practical perspective, we are interested in providing human annotators with a tool to assist them in adding semantic meta-data to normative documents.

2. Related Works

Our approach has some similarities with a number of previous works. The SALEM project has many similarities with our investigation [13,3]. SALEM automatically annotates the modificatory provisions of NIR documents by using syntactic parsing and a rule-based strategy to fill the semantic frames. In the SALEM project only a shallow syntactic analysis is produced, by a chunk parser. This system can be hardly compared to ours, in that a systematic experimentation -at the best of our knowledge- has not yet been carried on. However, the authors report a very high accuracy on, though on a limited test set.

The work described in [8] uses a deep syntactic parser (Collins' one [4]) to build a full syntactic description of legal sentences. Moreover McCarty uses a logic language, i.e. a deep semantic structure, to represent the entire semantics of the sentences, rather than focusing on semantic frames. The aim of the McCarty's work is to extract enough information to build a question answering system on judicial opinions, while the final aim of our project is the automatic annotation of the meta information regarding modificatory provisions.

Our system shares the same input representation and is based on the TUP parser, as well as the work in [6]. However, a big difference between the current investigation and the mentioned system is in the amount of syntactic information available throughout the

semantic annotation process. While Lesmo and colleagues do exploit dependency trees (that result from the parsing step), we are presently considering a simpler and by far less noisy text representation, only involving words order and PoS tags.

3. Documents and Provisions Encoding

This work relies on the *NormeInRete* standard (or NIR) for Italian Legal Text. NIR standard defines some structural elements that are used to mark up the main elements of a legal text, as well as its atomic parts (such as articles, paragraphs, *etc.*) and any non-structured text fragment.

A provision can be encoded through a specially defined space called ⟨*meta*⟩, in which a *URN* connects the element expressing a qualification with the textual element referred to (be this an atomic element, or a text string). We report here the modificatory provision model and definitions, presented in [11], on which our research is based. A *modificatory clause* includes the following information:

- *ActiveNorm*: the provision that states the normative modification;
- *PassiveNorm*: provision that is affected by the modification;
- *Action*: action produced by the active provision on the passive norm;
- *Times*: interval of enter in force of the modificatory provision and the interval of efficacy;
- *Content*: the part of the speech that models the old text to replace or repeal in the modified provision, as well as the new text is inserted in the destination;
- *Purview*: a part used to describe a modification, as by specifying any exception, extensions, or authorized interpretations;
- *Space*: a function used to specify a geographical area to which the modification applies;
- *Conditions*: where a modification is an effect dependent on an event, a geographic space, or a class (or domain) of application.

The NIR standard includes in its Document Type Definitions a part dedicated to modifications to implement this model in XML. Figure 1 illustrates how a non-qualified provision can be enriched with semantic metadata (bold formatted) by marking it up in XML through NormeInRete.

Semantic metadata are linked to structural elements by an URN to assert the kind of modification (action), the active and passive norms, and other sub-elements describing the action. Several classes are used to qualify the behavior of modificatory provisions: these classes are identified by the namespace *dsp:*. Every class of modificatory provisions is modeled as well by a number of sub-elements that further specify it.

In the example in Figure 1, the tag *dsp:substitution*, linked to the text of modificatory provision specifies the Action; the tag *dsp:norma*, linked to the structural element *rif* (normative reference) specifies the Passive Norm; the tag *novella*, linked to the structural element *virgolette* (quoted text) specifies that the Quoted Text should be added in the Active Norm; the tag *novellando*, linked to structural element *virgolette* specifies that the quoted text should be deleted in the Active Norm.

```
<dsp:sostituzione implicita="no">
  <dsp:pos xlink:href="#art1-com1-let38" xlink:type="simple"/>
  <dsp:norma xlink:href="..." xlink:type="simple">
    <dsp:pos xlink:href="#rif73" xlink:type="simple"/>
    <dsp:sub xlink:href="..." xlink:type="simple"/>
  </dsp:norma>
  <dsp:novella>
    <dsp:pos xlink:href="#mod68-vir1" xlink:type="simple"/>
  </dsp:novella>
</dsp:sostituzione>

<el id="art1-com1-let38" xml:lang="it">
 <num>38.</num>
 <corpo>L' <mod id="mod68" xml:lang="it"><rif id="rif73" xlink:href="..."
   xlink:type="simple" xml:lang="it"> articolo 92</rif> soppresso dal testo
   seguente: <virgolette id="mod68-vir1" tipo="struttura" xml:lang="it">
   "<articolo id="mod68-vir1-art92" xml:lang="it"><num>Articolo 92</num>
   <rubrica xml:lang="it">Compilazione del rapporto di ricerca europea</rubrica>
   <comma id="mod68-vir1-art92-com1" xml:lang="it"><num/><corpo>L'Ufficio europeo
   dei brevetti redige e pubblica, in conformità al regolamento di esecuzione, un
   rapporto di ricerca europeo relativo alla domanda di brevetto europeo sulla
   base delle rivendicazioni, in debita considerazione della descrizione e, se
   del caso, dei disegni esistenti.</corpo></comma></articolo></virgolette></mod>
 </corpo>
</el>
```

Figure 1. A substitution provision with structural and semantic markup for the Italian phrase *L' articolo 92 è soppresso dal testo seguente: L'Ufficio europeo dei brevetti redige e pubblica, in conformità al regolamento di esecuzione, un rapporto di ricerca europeo relativo alla domanda di brevetto europeo sulla base delle rivendicazioni, in debita considerazione della descrizione e, se del caso, dei disegni esistenti.*

4. The Eunomos system

Eunomos [2] is a legal knowledge management service which includes the ability to view legislation from various sources and find definitions and explanations of legal concepts by legal experts. It was developed as part of the ICT4LAW[1] project, funded by Regione Piemonte, and distributed by Nomotika, a commercial spinoff of the University of Turin.

Currently Eunomos can find most explicit references using the XML Leges Linker. A subsequent module is then in charge of determining whether the reference is a simple reference or whether it modifies or overrides other legislation. The system in [6] was an initial prototype achieving that task. The present paper presents a new prototype.

As pointed out in the Introduction, the crucial difference between the new system and the one in [6] is that the former does not make use of TUP dependency relations. In fact, we observed in the results of the previous system that most errors were produced by such syntactic relations. In fact, the TUP parser was designed to be a multi-purpose parser, i.e. it was designed to parse any kind of NL text, not necessarily legal one. As a consequence, parsing *specific* text intrinsically produces some errors due to the (multi-purpose) rules inserted in TUP for handling *general* text.

The system presented here exploits only TUP morphological analysis of the input words and defines a set of ad-hoc rules that consider such analysis as well as words order. In practice, the ad-hoc rules implement a new (shallow) parser specifically dedicated to recognize the text expressions used by the modificatory provisions. Moreover, the system design easily allows to achieve other two advantages:

[1] http://www.ict4law.org/

(1) a. Rules may be prioritized, so that more than one rule is usually triggered at the same time, but only the rule with higher priority is actually executed. This allows to add rules without modifying the previous ones. Whether a provision is not processed, or it is not processed in the proper way, rather than debugging the set of rules already defined in the system, we simply add a new one with *higher priority* that produces the right annotation for the provision at hand. As a positive consequence, it is rather simple to iteratively update the system to get higher levels of recall and precision[2].

b. It is rather simple to implement procedures that keep track of the effects of the new rules. In particular, we implemented procedures that, if a new rule is introduced, process the corpus both with the old system and the new one. In this way, it is possible to detect side effects of the new rule, i.e. to check whether the system updated is no longer able to process provisions that were properly annotated in the previous version. In such a case, we may then debug the new rule and modify it so that it gets able to annotate new provisions without preventing the annotation of the ones previously identified. This "double-check" guarantees incrementality in the growth of the system: the more provisions are analyzed, the more exhaustive and precise the set of rules becomes.

In other words, (1.a-b) have been implemented in order to (semi-automatically) assist and control the writing of the rules. (1.a-b) guarantees *incrementality* in the sense that once a new rule is added to the system's knowledge base, (1.b) detects if the new rule interferes with other rules, so that it is possible to assign a lower priority to the new one. Thus, an higher recall is obtained without worsening the precision.

5. A set of shallow parsing rules for classifying modificatory provisions

The rules are implemented in XML, but space constraints prevent us from illustrating in detail the XML format adopted. The rules takes in input the explicit references identified by the XML Leges Linker, and the TUP morphological analysis of the legal text.
The general pattern of the rules is illustrated in Fig.2.

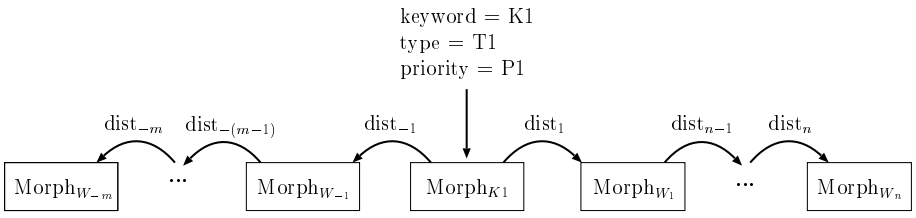

Figure 2. General pattern of the system rules

[2]On the other hand, as a negative consequence, as long as the system is updated time performances get worse. In fact, the set of rules only grows, as rules are never deleted, so that the average number of rules that are processed on the same portions of text becomes progressively larger. We could implement procedures that identify and prune useless rules, i.e. rules that never lead to an annotation as it is always the case that some other rules, with higher priority, triggers on the same text, but we do not implement such procedures yet as we are not interested, at the moment, at keeping the time performances as lower as possible.

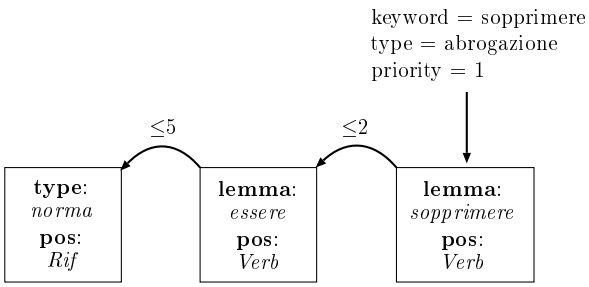

Figure 3. A rule for certain kind of 'abrogazioni' (*abrogations*)

The system scans the words in the input text and, in case it finds a word with lemma *K1*, it triggers the rule in fig.2. Then it carries out three checks.
Firstly, it checks if the morphological information of the keyword with lemma *K1* match the one in $Morph_{K1}$. Then it checks if the words that follow the keyword in the words order respectively match the morphological descriptions $Morph_{W_{n1}}, \ldots, Morph_{W_{nx}}$ and if the words that precede it respectively match the morphological descriptions $Morph_{W_{p1}}, \ldots, Morph_{W_{py}}$.
$dist_{n1}, \ldots, dist_{nx}, dist_{p1}, \ldots, dist_{py}$ are integer specifying the maximal distance among a pair of words. For instance, between the keyword and the word W_{n1} there could be $dist_{n1}$ other words.
In case the three checks are satisfied, the provision is classified as type *T1*. Moreover, among the words $W_{n1}, \ldots, W_{nx}, W_{p1}, \ldots, W_{py}$, there could be some normative references, that could be classified by the rule as either *norma*, *novella*, or *posizione*. In case the constraints specified in the rule are satisfied, the final annotation will specify the normative references recognized by it.

Fig. 3 shows an example of instantiation of the pattern in Fig. 2. The rule triggers in case the system finds in the input text a verb with lemma 'sopprimere' (*to suppress*). Then, it checks if, among the two[3] preceding words, there is a verb with lemma 'essere' (*to be*), and if, among the five preceding words of the latter there is a normative reference. In case this is true, the provision is annotated as 'abrogazione', with the normative reference occurring therein identified as 'norma'.
Many provisions are correctly classified by the rule in Fig. 3. Nevertheless, the rule also leads to wrong annotations. For instance, it triggers also for the provision shown in Fig. 1, wrongly classifying it as 'abrogazione'. Although the main verb of the provision in Fig. 1 is 'abrogare', that text is a 'sostituzione'. More in general, the sentences in the form 'Il rif1 è abrogato da rif2' (*The rif1 is abrogated by rif2*) are substitutions, not abrogations. Therefore, we add in the system the rule in Fig.4, and we of course assign it a priority higher than the one of the rule in Fig. 3.
The checks carried out on the words preceding the keyword 'abrogare' are the same of those in fig.3. Furthermore, the rule in fig.4 requires the occurrence of the preposition 'da' immediatly after the keyword and a normative reference (that will be annotated as

[3]We specified a maximum distance of 2 words in order to encompass both sentences in the form 'Il rif1 è abrogato' (*The rif1 is abrogated*) and sentences in the form 'Il rif1 è stato abrogato' (*The rif1 has been abrogated*). In Italian, the lemma of both words 'è' and 'stato' is 'essere'.

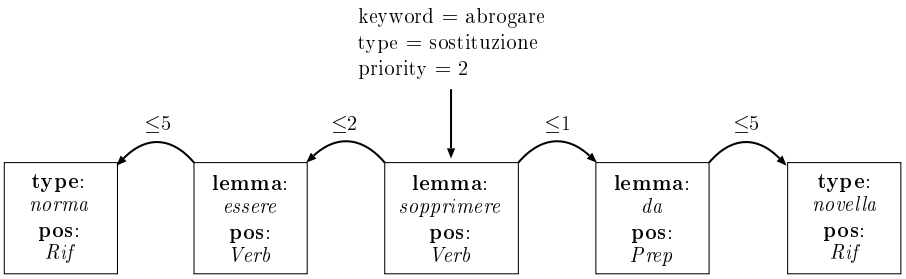

Figure 4. A rule for certain kind of 'sostituzioni' (*substitutions*)

'novella') among the five words following the preposition. Note that, with respect the provision in fig.1, the 'novella' is the fourth word following the keyword.

5.1. Coordinations

It is worth mentioning that the format of the XML rules is not suitable for managing coordinations. In fact, that format is rather static, while coordinations feature a *dynamic* number of conjuncts. For instance, consider the example in (2), where `ref1`, `ref2`, `ref3`, and `ref4` are references to suppressed articles, identified by the XML Leges Linker.

(2) Gli articoli: `ref1`, `ref2`, `ref3`, e `ref4` sono soppressi.
(The articles: `ref1`, `ref2`, `ref3`, and `ref4` are suppressed.)

In (2), there are four articles. However, of course an arbitrary number of articles could be conjoined in the subject of the sentence.
In order to detect all references involved in a coordinations, we introduced *recursive* rules. For instance, fig.5 defines the rule for properly dealing with coordinations as the one exemplified in (2). `REC1` is a substructure/subrule that is (recursively) executed everytime a *norma* is identified. Thus, the rule(s) in fig.5 are able to recognize a set of references with dynamic size.

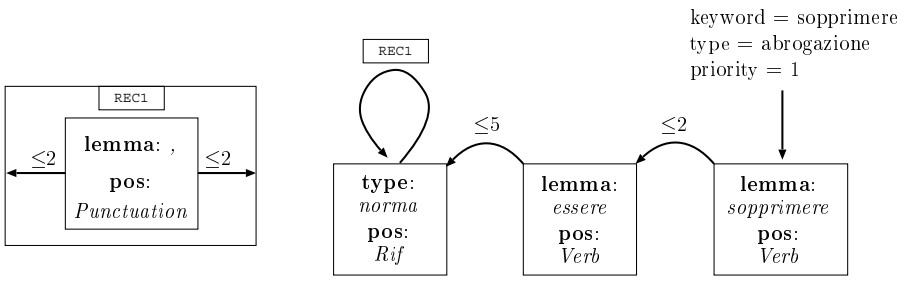

Figure 5. A rule for certain kind of 'sostituzioni' (*substitutions*)

The example in (2) is rather simple, so that it is still easy to extend the formalism in order to handle it. However, of course the corpus includes cases of coordinations much more complex than this. The rules defined for handling such cases are rather complex, and their effects on the other rules hard to control. We stress again that the reason is that

the adopted formalism is *intrinsically* not suitable for managing dynamic structures. On the other hand, the dependency relations identified by the TUP parser, which is able to deal with recursive linguistic expressions like coordinations, appears to be a promising solution for that specific task. The definition of an hybrid approach able to exploit the advantages of both the present prototype and the one in [6] is seen as the object of future works.

6. Experimentation

Our system has been evaluated on the same test set used by the system in [6]. The results of both are shown in Table 1.

Table 1. Evaluation of the system, and comparison with [6]

	Recall	**Precision**
[6]'s System	71.7%	83.0%
Current System	86.60	98.56

These results provide a clear improvement over a previous experimentation reported in the work by Lesmo & colleagues [6]. We defer to future works a systematic comparison between current system and the cited one and presently outline few aspects that deserve being mentioned.

However one major strength of current system is in its simplicity, compared with the multi-layered architecture by [6]; it is thus easier to handle and tune it with respect to the multi-layered architecture. Also, since it accomplishes a task that is by far simpler –also on a computational perspective– it is faster, as well as robust. On the other hand, the updating process, i.e. the task of adding new rules to the system, is quite slow and boring, although the "double-check" procedures described in (1.b) above at least guarantees their incrementality, i.e. that the output of the rules become progressively more accurate.

Finally, it is worth noticing that the system presented here achieves an higher level of precision, close to 100%, because the rules behave as a kind of "filter". In other words, the system uses *ad-hoc* rules, each of them describes a valid pattern. As a consequence, (almost) any provision matching with this pattern is precisely classified by the pattern itself. On the contrary, in the system by [6], the errors done by the TULE parser, which is a *multi-purpose* parser, propagate in the final result. However, a deeper analysis is needed to fully assess the system and to compare it to the full-parsing approach. In fact, the multi-layered architecture by [6] (using a general-purpose parser) would apparently seem to be more general. It relies on same tens of 'wide-coverage' rules that are used to describe a plethora of different linguistic realizations, e.g. coordinations, which are responsible of some of the missed recall of our system.

7. Conclusions

Information technology is a natural ally for legal research, characterised as it is by constant cross-referencing, updates and obscure terminology. Natural language processing is essential for efficient semantic analysis of legislative text.

In this paper we illustrate ongoing work on the Eunomos system. In particular, we presented a tool for classifying types of modifications in Italian legal text, according to the *NormeInRete* (NIR) standard. The design of the tool was based on the analysis of the errors in the output of a previous prototype [6]. Basically, the majority of errors were due to the dependency relations identified by the TUP parser, that is a multi-purpose parser for Italian. The new module considers only the morphological analysis of the TUP parser, while it ignores its syntactic analysis. Then, it defines a new set of XML shallow parsing rules for identifying the type of modifications. The performances are much higher than the previous tool, especially the precision, because the rules act as a kind of "filter".

Concerning our future works, we plan to use the tool for recognising types of modifications presented here in a new module for automatically generating different versions of consolidated text, as done by Palmirani e Brighi [9]. Currently the Eunomos system stores the original and most recent versions of legislation, and this is sufficient for the needs of prospective users. Nevertheless, the Eunomos system contains a functionality for adding any number of intermediate versions, so a consolidation module could be added in the future if required.

References

[1] C. Biagioli, E. Francesconi, P. Spinosa, and M. Taddei. The NIR project: Standards and tools for legislative drafting and legal document web publication. In *Procs. of ICAIL Workshop*, pages 69–78, 2003.

[2] G. Boella, L. Humphreys, M. Martin, P. Rossi, and L. van der Torre. Eunomos, a legal document and knowledge management system to build legal services. In *AI Approaches to the Complexity of Legal Systems (AICOL11)*, 2011.

[3] M. Cherubini, G. Giardiello, S. Marchi, S. Montemagni, P. Spinosa, and G. Venturi. NLP-based metadata annotation of textual amendments. In *Proc. of WORKSHOP ON LEGISLATIVE XML 2008, JURIX 2008*, 2008.

[4] M. Collins. Three generative, lexicalised models for statistical parsing. In *In Proc. of the 35th Annual Meeting of ACL*, pages 16–23, 1997.

[5] L. Lesmo. The Rule-Based Parser of the NLP Group of the University of Torino. *Intelligenza Artificiale*, 2(4):46–47, June 2007.

[6] L. Lesmo, A. Mazzei, and D.P. Radicioni. Extracting Semantic Annotations from Legal Texts. In *Procs. of HT09*, pages 167–172, Turin, Italy, July 2009. ACM.

[7] C. Lupo, F. Vitali, E. Francesconi, M. Palmirani, R. Winkels, E. de Maat, A. Boer, and P. Mascellani. General XML format(s) for legal Sources - ESTRELLA European Project. Deliverable 3.1, Faculty of Law, University of Amsterdam, Amsterdam, The Netherlands, 2007.

[8] L. T. McCarty. Deep semantic interpretations of legal texts. In *Proc. of ICAIL Workshop*, pages 217–224, New York, NY, USA, 2007. ACM.

[9] M. Palmirani and R. Brighi. Norma-system: A legal document system for managing consolidated acts. In *Database and Expert Systems Applications, 13th International Conference, DEXA 2002, Proceedings*, volume 2453 of *Lecture Notes in Computer Science*, pages 310–320. Springer, 2002.

[10] M. Palmirani and R. Brighi. An XML Editor for Legal Information Management. In *Electronic Government*, LNCS, pages 421–429. Springer, 2003.

[11] M. Palmirani and R. Brighi. Time Model for Managing the Dynamic of Normative System. *Electronic Government*, pages 207–218, 2006.

[12] J. Saias and P. Quaresma. A Methodology to Create Legal Ontologies in a Logic Programming Based Web Information Retrieval System. *Artificial Intelligence and Law*, 12(4):397–417, 2004.

[13] C. Soria, R. Bartolini, A. Lenci, S. Montemagni, and V. Pirrelli. Automatic Extraction of Semantics in Law Documents. In C. Biagioli, E. Francesconi, and G. Sartor, editors, *Proceedings of the V Legislative XML Workshop*, pages 253–266. European Press Academic Publishing, 2007.